## CLAIMS

We claim:

1. A method for controlling access to an object in an operating system, the method comprising:

receiving a call from an external object to a first interface of a target object;

at the target object, determining whether the external object has access to other interfaces of the target object based on the call to the first interface; and

granting access to the other interfaces according to the determination.

2. A method as recited in claim 1, wherein determining whether the external object has access to other interfaces of the target object further comprises examining a security policy contained within the target object.

3. A method as recited in claim 2, wherein the security policy is contained entirely within the target object.

4. A method as recited in claim 1, further comprising determining whether the external object and the target object operate in the same process.

5. A method as recited in claim 1, wherein determining whether the external object has access to other interfaces of the target object further comprises:

identifying other interfaces of the target object that can be accessed when the first interface is being requested by the external object.

6. A method as recited in claim 1, further comprising determining a first process of the target object.

7. A method as recited in claim 6, further comprising determining a second process of the external object.

8. A method as recited in claim 7, further comprising performing a cross-process communication between the target object and the external object.

9. A method as recited in claim 1, further comprising securing a channel for each interface of the target object.

10. A method as recited in claim 1, wherein determining whether the external object has access to other interfaces of the target object further comprises analyzing access constraints within the target object.

11. A method as recited in claim 1, further comprising analyzing interface access data stored within the target object.

12. A method as recited in claim 1, further comprising determining whether the target object and the external object are in a same protection domain.

13. A method as recited in claim 12, wherein the protection domain is a process.

14. A method as recited in claim 1, wherein the target object sets its own security policy.

15. A method as recited in claim 1, wherein determining whether the external object has access to other interfaces further comprises determining the capabilities of the external object.

15. A method as recited in claim 14, further comprising mapping the capabilities of the external object to the interfaces of the target object.

16. A method as recited in claim 1, wherein the target object and the external object are created using a same methodology.

17. A method as recited in claim 1, wherein the target object and the external object are views in a view hierarchy.

18. A method as recited in claim 17, wherein a view has a parent calling interface, a child calling interface, and a child managing interface.

19. A system that controls access to an object in an operating system, the system comprising:

a module configured to receive a call from an external object to a first interface of a target object;

a module configured to determining whether the external object has access to other interfaces of the target object based on the call received at the first interface; and a module configured to grant access to the other interfaces according to the determination.

20. A system that controls access to an object in an operating system, the system comprising:

means for receiving a call from an external object to a first interface of a target object;

means for determining, at the target object, whether the external object has access to other interfaces of the target object based on the call to the first interface; and

means for granting access to the other interfaces according to the determination.

21. A computer readable medium storing instructions for controlling a computer device to control access to an object in an operating system, the instructions comprising:

receiving a call from an external object to a first interface of a target object;

at the target object, determining whether the external object has access to other interfaces of the target object based on the call to the first interface; and

granting access to the other interfaces according to the determination.

22. A method for securing an object in a computing device operating system, the method comprising:

determining one or more access constraints of a first object;

identifying a protection domain that has a security profile that corresponds to the one or more access constraints of the first object; and

placing the first object in the protection domain.

23. A method as recited in claim 22, further comprising creating the first object and a second object using the same methodology.

24. A method as recited in claim 23, wherein the first object and the second object can communicate transparently across two or more protection domains.

25. A method as recited in claim 22, wherein the protection domain is a process.

26. A method as recited in claim 22, further comprising creating an object-to-object security model wherein security constraints for an object are contained within the object.

27. A method as recited in claim 22, wherein identifying a protection domain further comprises attempting to identify a protection domain that is local relative to the first object.

28. A method as recited in claim 22, further comprising creating a process based on security requirements of the operating system.

28. A method as recited in claim 28, further comprising clustering objects in the process based on security policies of the objects.

29. A system for securing an object in a computing device operating system, the system comprising:

means for determining one or more access constraints of a first object;

means for identifying a protection domain that has a security profile that corresponds to the one or more access constraints of the first object; and

means for placing the first object in the protection domain.